客户端健壮性测试工具 Looper

郭舜

美团点评

美团平台及酒旅事业群 | 酒旅测试组

郭舜

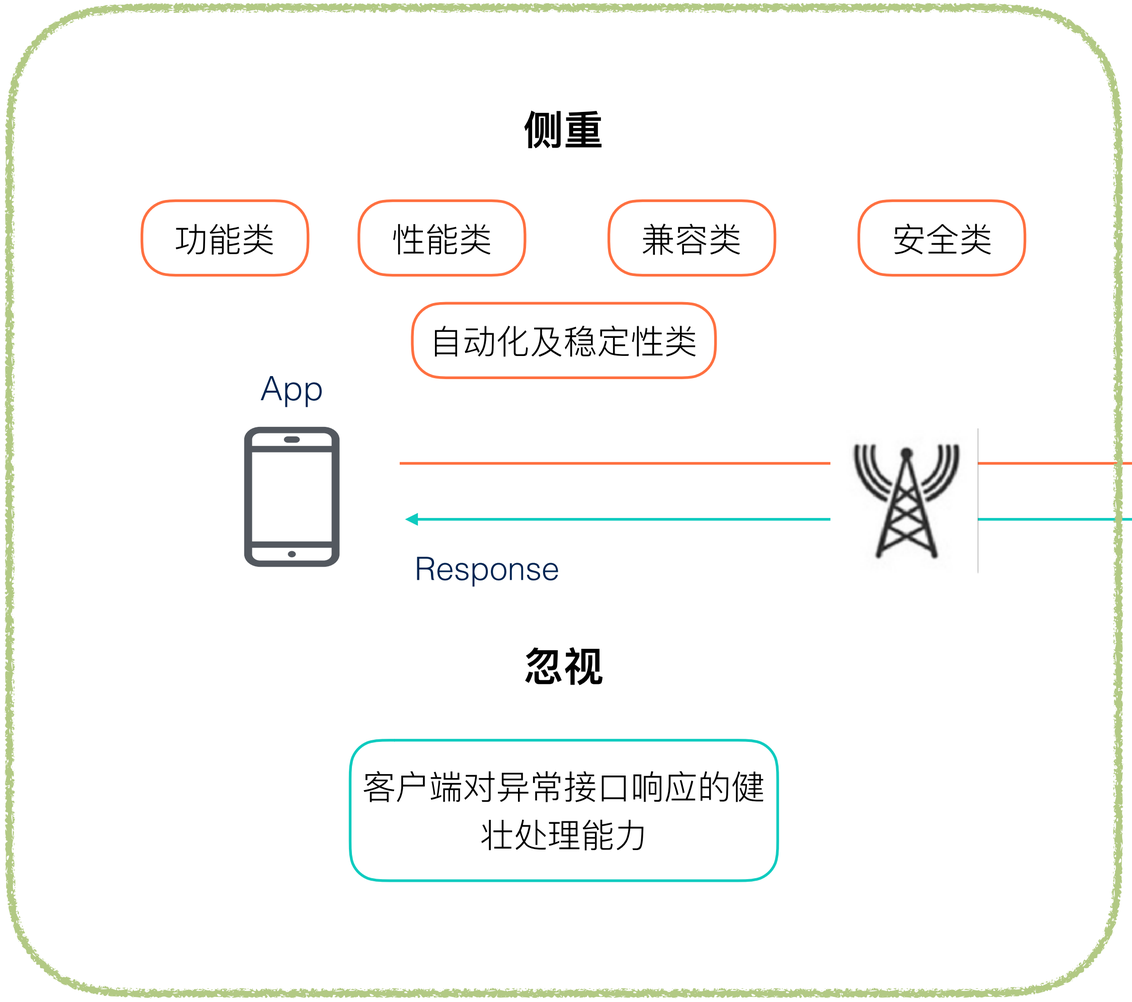
美团酒旅高级测试开发工程师

移动端测试方向

2014年加入美团，一直从事美团App的客户端质量保证工作，主要研究客户端自动化、集成测试以及专项测试方面，旨在尽可能地深入细节，更全面的保证App质量

目录

App测试专项盲区



App能完全相信后台吗?

App对数据足够健壮吗?

血的教训

A

页面 /Home /技术研发部 /Hotel /酒店项目组-CaseStudy /CaseStudy-2017

CaseStudy-20170515-[用户后台B组]点评酒店IOS客户端Crash

三、故障原因

本次问题产生的原因，原来的字段是String类型，切换新的服务后变成double类型。

```
//add real time comment & score
Map<Integer, HotelShopCommentDTO> dtoMap= new QueryShopCommentTagInfoCommand(shopId).execute();
if(null != dtoMap && dtoMap.size() != 0){
    HotelShopCommentDTO dto = dtoMap.get(shopId);
    if(null != dto) {
        cachedHotelInfo.put("score", dto.getScore());
        cachedHotelInfo.put("scoreDesc", dto.getScoreText() == null ? "" : dto.getScoreText());
    }else{
        cachedHotelInfo.put("score", "");
        cachedHotelInfo.put("scoreDesc", "");
    }
}else{
    cachedHotelInfo.put("score", "");
    cachedHotelInfo.put("scoreDesc", "");
}
```

解决方案

评论只是商家的一个小模块，但是它的接口微调却引起了客户端商家详情页面的Crash，导致主流程Block

B

页面 /... /线上事故总结 - Android

caseStudy-Android美团7.0海外列表页选择景点类目稳定crash

原因分析及解决方法

原因分析

```
for (int i = 0; i < poi.payAbstracts.size(); i++) {
    OverseaDealItemView dealView = new OverseaDealItemView(context);
    dealView.setIcon(poi.payAbstracts.get(i).icon_url);
    dealView.setContent(poi.payAbstracts.get(i).title);
    dealLayout.addView(dealView);
}
```

使用payAbstracts接收交易信息，直接使用list.size迭行遍历，进行UI拼装，未进行空判断。而后台返回数据为

```
"payAbstracts": [
  null,
  {
    "abstract": "1749元浅草寺+仲见世通+筑地市场+皇居一日游,1600元浅草寺+仲见世通+筑地市场+皇居一日游",
    "icon_url": "http://p1.meituan.net/mmc/1395bdb2d62af1b10355be8857eb168c2359.png",
    "type": "travel"
  }
],
```

数组中，第一个值为null，第二个值有效，导致list的size为2，但是item只有一个数据，发生空指针崩溃。

在正常的数据集下面App是能正常处理的，但是后台数据集异常的时候客户端就直接Crash了

健壮性问题

01 很多异常数据集和状态会直接影响线上客户端Crash率以及线上的正常流程

02 互联网公司业务逻辑越来越复杂，上线越来越频繁，极大提高了线上App的数据兼容风险

03 人工的App异常和边界测试总会有不全面的时候，如何去及时暴露出客户端健壮性问题?

客户端健壮性挑战与目标

- App打铁还需自身硬，默认不信任后台，做到自我健壮

- App能应对后台的异常情况，不Crash是底线，不会因为小模块的数据异常引起主流程block。

目录

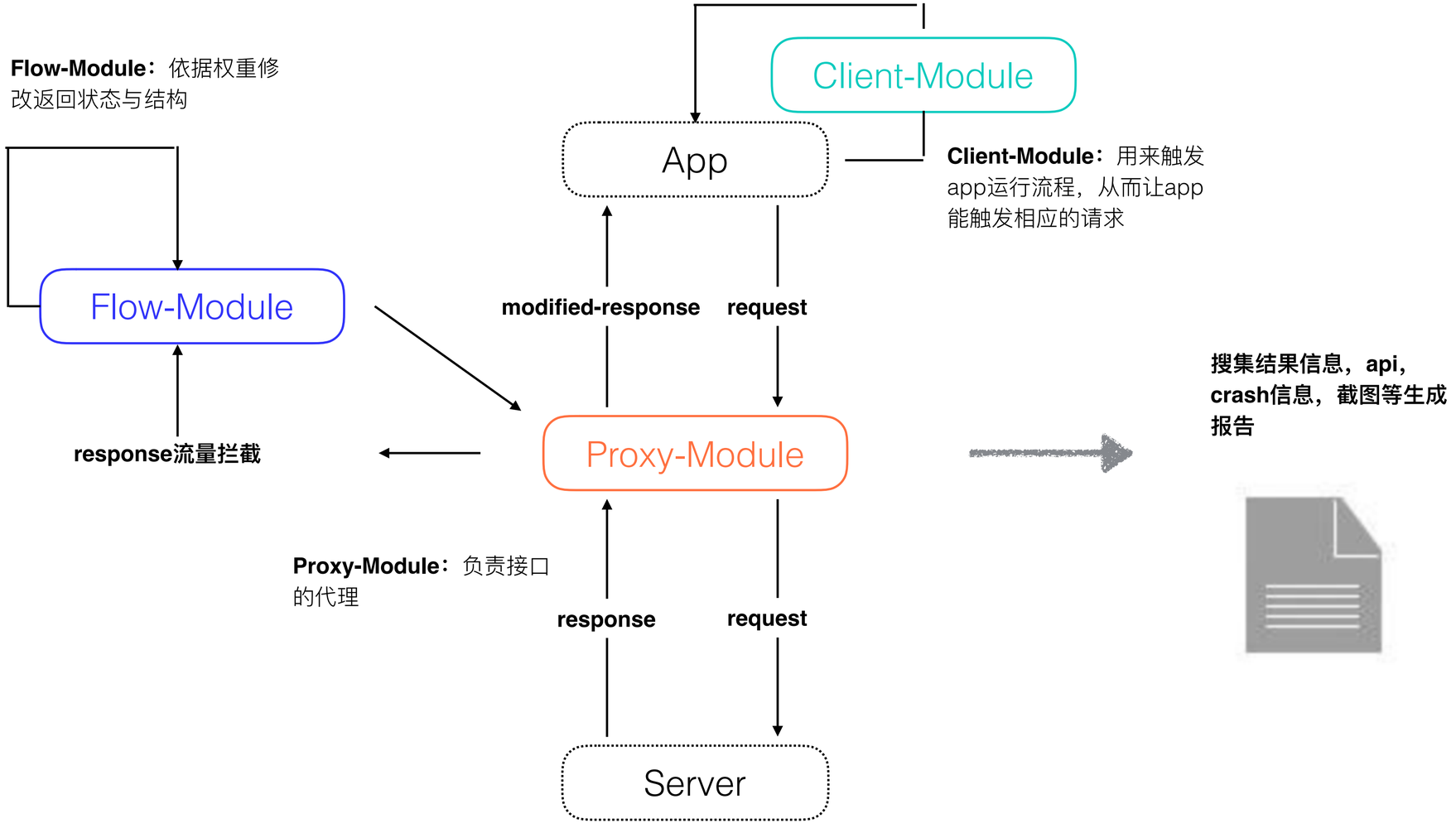
解决方案

提高业务QA对接口异常场景与异常数据测试的覆盖

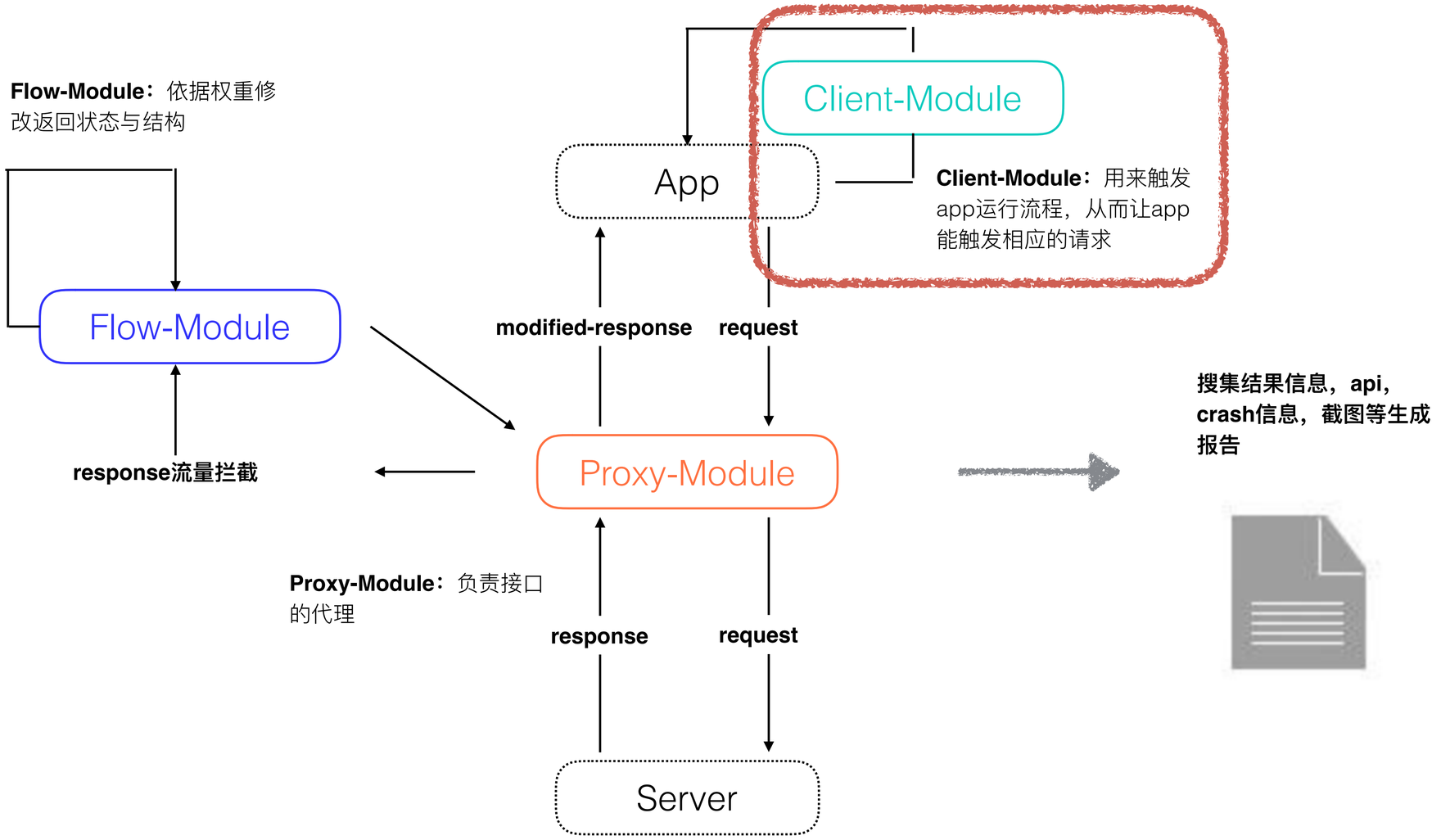
开发一个工具，在整个测试研发阶段不断地对App页面进行接口随机破坏，尽可能将潜在App健壮性问题暴露出来

Looper

Looper整体框架


Client-Module
App
Client-Module：用来触发app运行流程，从而让app能触发相应的请求
Flow-Module：依据权重修改返回状态与结构
Flow-Module
modified-response
request
response流量拦截
Proxy-Module
搜集结果信息，api，crash信息，截图等生成报告
Proxy-Module：负责接口的代理
response
request
Server

Looper整体框架


Flow-Module：依据权重修改返回状态与结构
Client-Module
App
Client-Module：用来触发app运行流程，从而让app能触发相应的请求
Flow-Module
modified-response
request
response流量拦截
Proxy-Module
搜集结果信息，api，crash信息，截图等生成报告
Proxy-Module：负责接口的代理
response
request
Server

Client-Module

选型对比

	优势	劣势	选择
Monkey	•简单易用	•轨迹记录麻烦 •无序，截图bug重放不适用 • 需要android与iOS各搞一套	✗
UI自动化	•触发流程完全可控 •可以精确log	•Case维护成本大而且复杂 •页面前后文依赖严重，深入页面不一定能覆盖 •不够稳定	✗
Schema跳转（单页面跳转）	•更有针对性，可以着重测试用户场景的主要页面 •iOS和android的策略和方案统一 •配置简单，无需上下文概念，而且回放Crash容易	•页面跳转性和参数可能需要Rd支持	✓

Schema跳转实现

旅游首页： imeituan://www.meituan.com/trip/homepage?selectedCityId=1

一般开发会将Android 与 iOS同一页面的跳转协议规定成一样

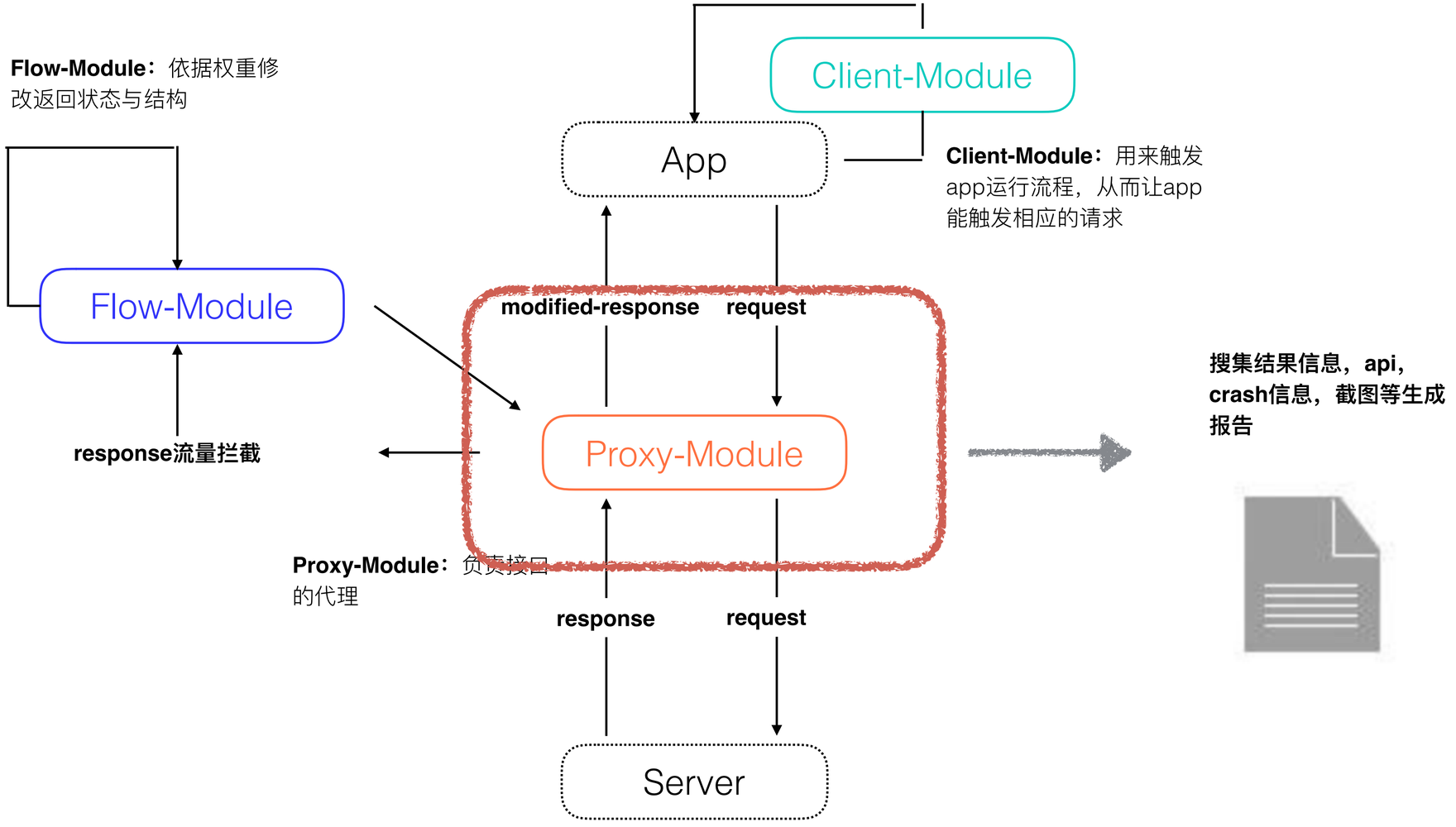
Android跳转：

- 直接用系统am命令发intent开启页面
- 参数拼接的&符号需要转译

iOS跳转：

- 利用Safari为跳板，直接填入相应的schema进行跳转

Looper整体框架


Flow-Module：依据权重修改返回状态与结构
Client-Module
App
Client-Module：用来触发app运行流程，从而让app能触发相应的请求
Flow-Module
modified-response
request
response流量拦截
Proxy-Module
搜集结果信息，api，crash信息，截图等生成报告
Proxy-Module：负责接口的代理
response
request
Server

Proxy-Module

直接用开源工具-Mitmproxy

维护者和社区活跃，功能Api丰富

请求控制

截取流量，过滤非JSON接口，并格式化body数据

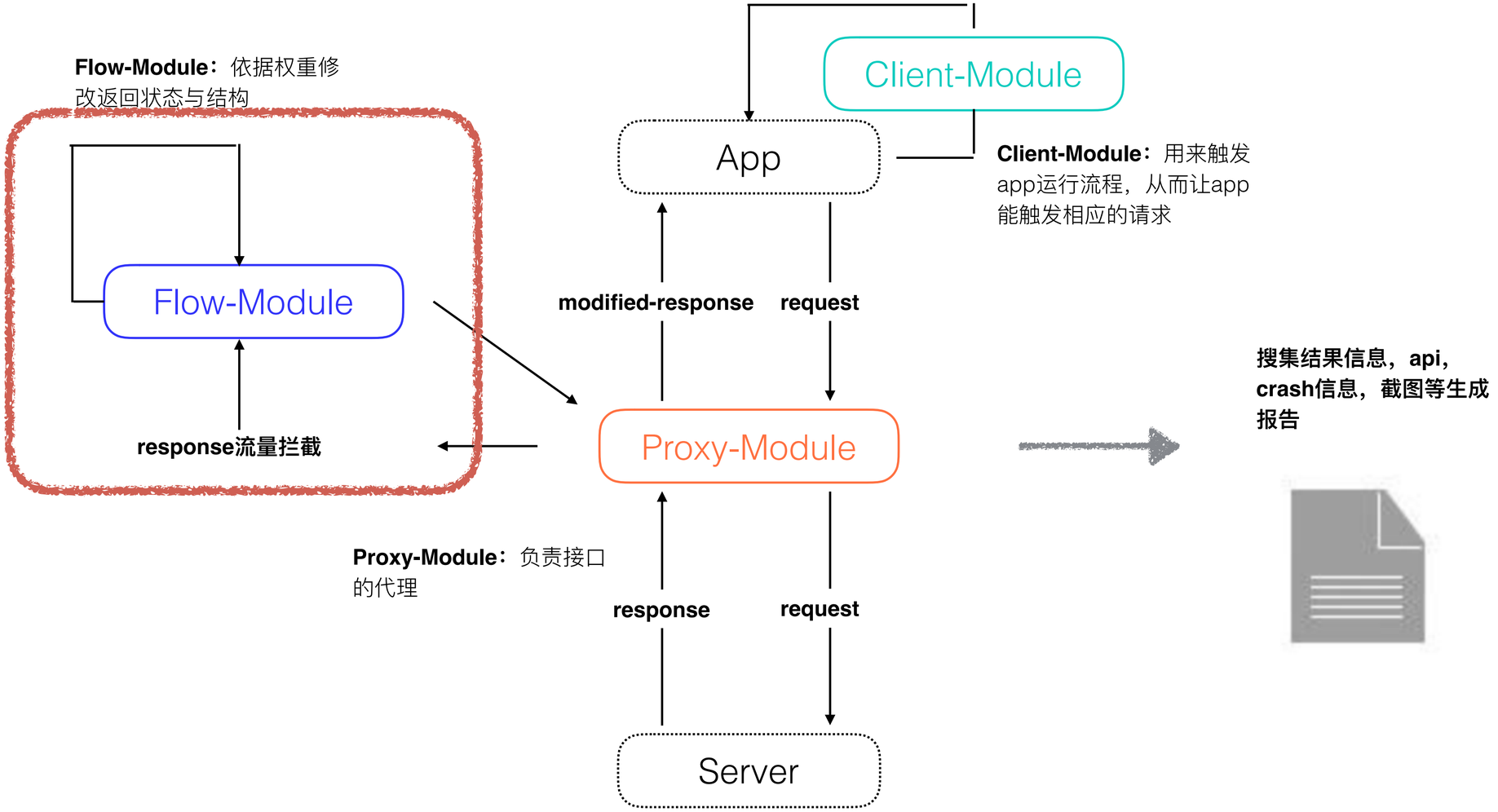
逻辑控制

分页面清空流量队列，保证单页面请求的唯一性

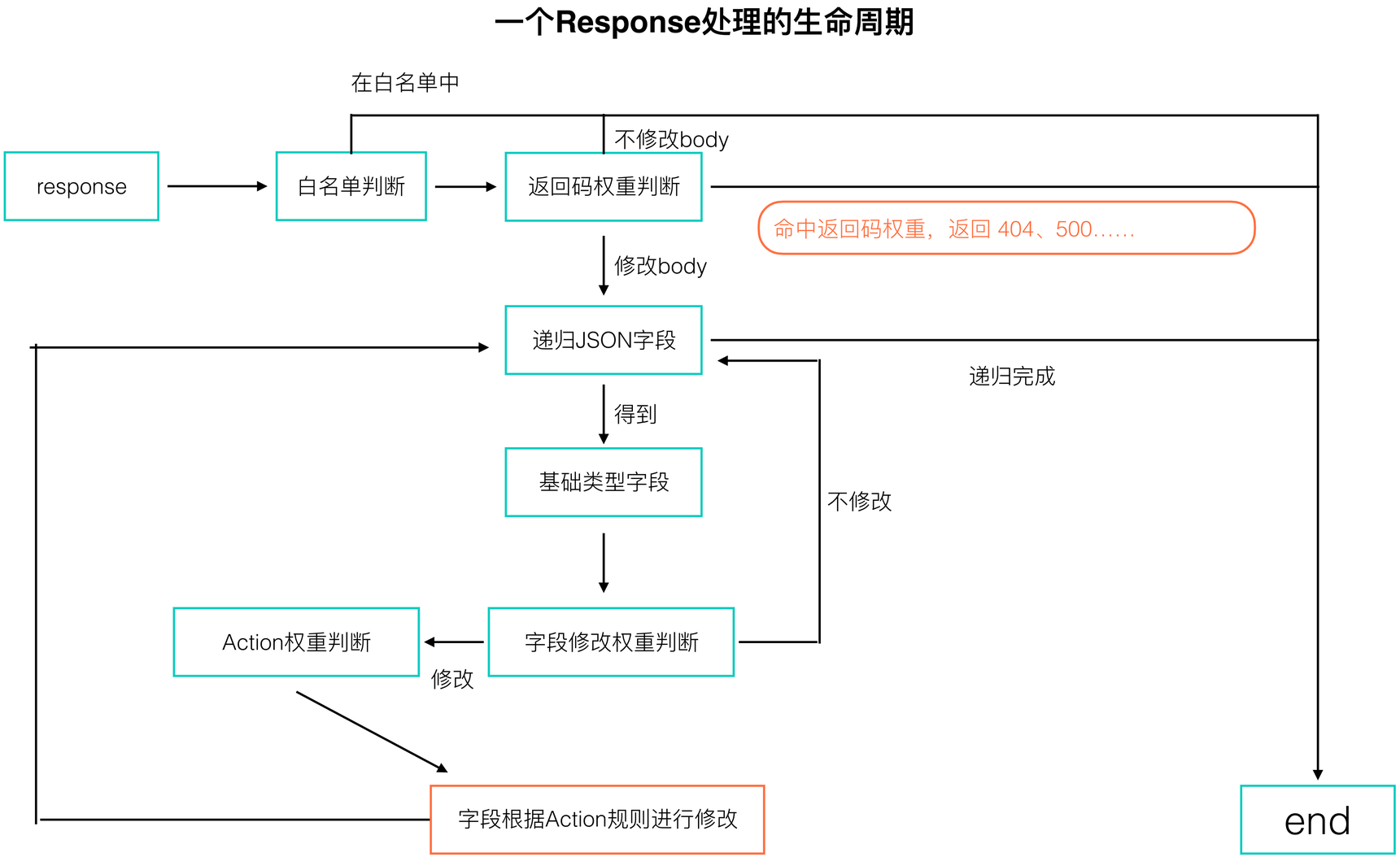
流量保存

每个页面的修改JSON与修改后的JSON都保存本地，方便回放

Looper整体框架


Flow-Module：依据权重修改返回状态与结构
Client-Module
App
Client-Module：用来触发app运行流程，从而让app能触发相应的请求
Flow-Module
modified-response
request
response流量拦截
Proxy-Module
搜集结果信息，api，crash信息，截图等生成报告
Proxy-Module：负责接口的代理
response
request
Server

Flow-Module


一个Response处理的生命周期
在白名单中
不修改body
response
白名单判断
返回码权重判断
命中返回码权重，返回 404、500……
修改body
递归JSON字段
递归完成
得到
基础类型字段
不修改
Action权重判断
字段修改权重判断
修改
字段根据Action规则进行修改
end

Flow-Module 权重设置

Resp生命周期三个权重判断的Hook

01 网络返回码位置

02 字段类型判断位置

03 ACTION规则选择位置

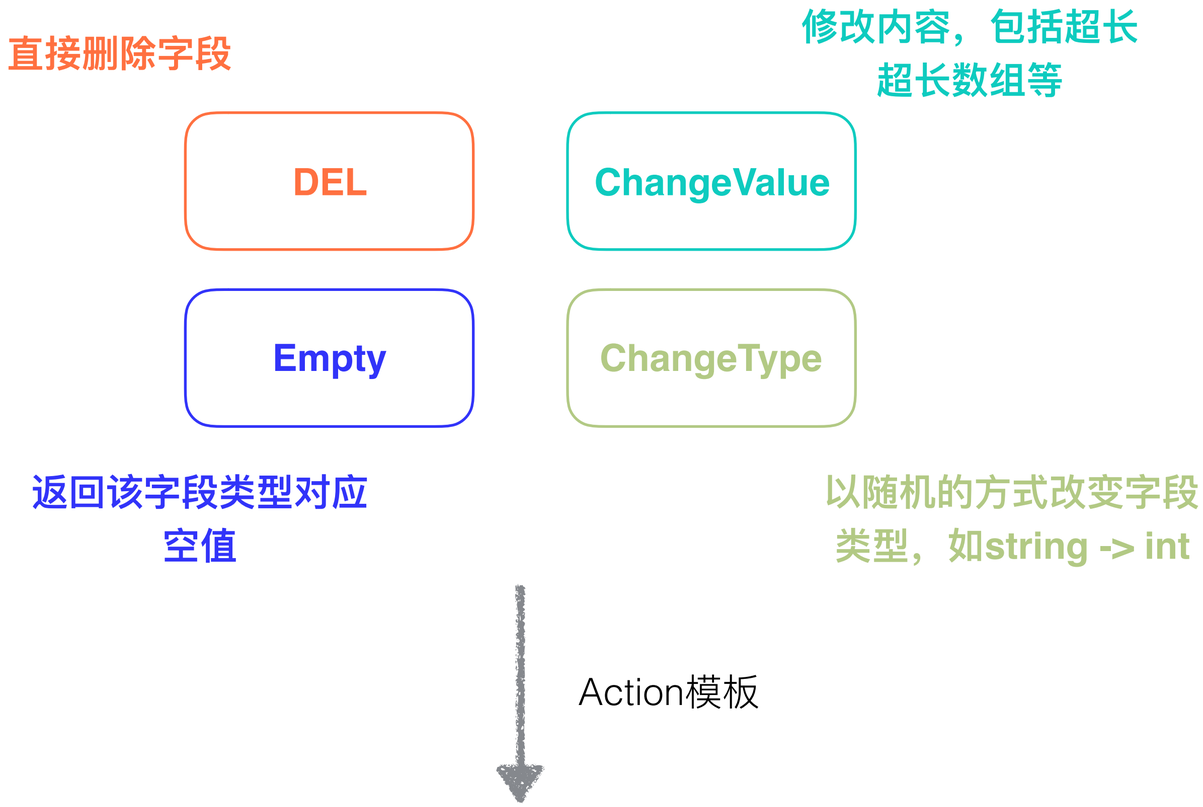
权重方案：

key-value数字百分比配置

```
ACTION_WEIGHT = {
    Operation.DEL: 10,
    Operation.CHANGE_TYPE: 10,
    Operation.CHANGE_VALUE: 30,
    Operation.EMPTY: 50
}
```

以**不完全随机**的方式进行接口修改，做到一定程度上的可控。可以分析结果，在周期运行中动态改变设置权重

Flow-Module Actions配置



支持扩展

Looper-详细的报告

schema具体信息

单schema每次跳转的截图和接口数据

详细的Crash分析日志与日志下载

Looper-场景回放功能

开发人员看到堆栈无法定位原因?

开发人员想要重现UI或者Crash

提供异常场景回放功能

- 大家可以根据结果，回放场景UI和Crash，便于开发人员更好的定位问题，只需要一行命令就可以重现异常场景

```
python runner.py replay --platform android \
    --sn 05157df5f1b4330e --package com.sankuai.meituan \
    --replay_dir /Users/guoshun/PycharmProjects/Robust/result_robust/0_homepage/1_data \
    --schema 'imeituan://www.meituan.com/trip/homepage?selectedCityId=1' \
    --port 9000'
'''
```

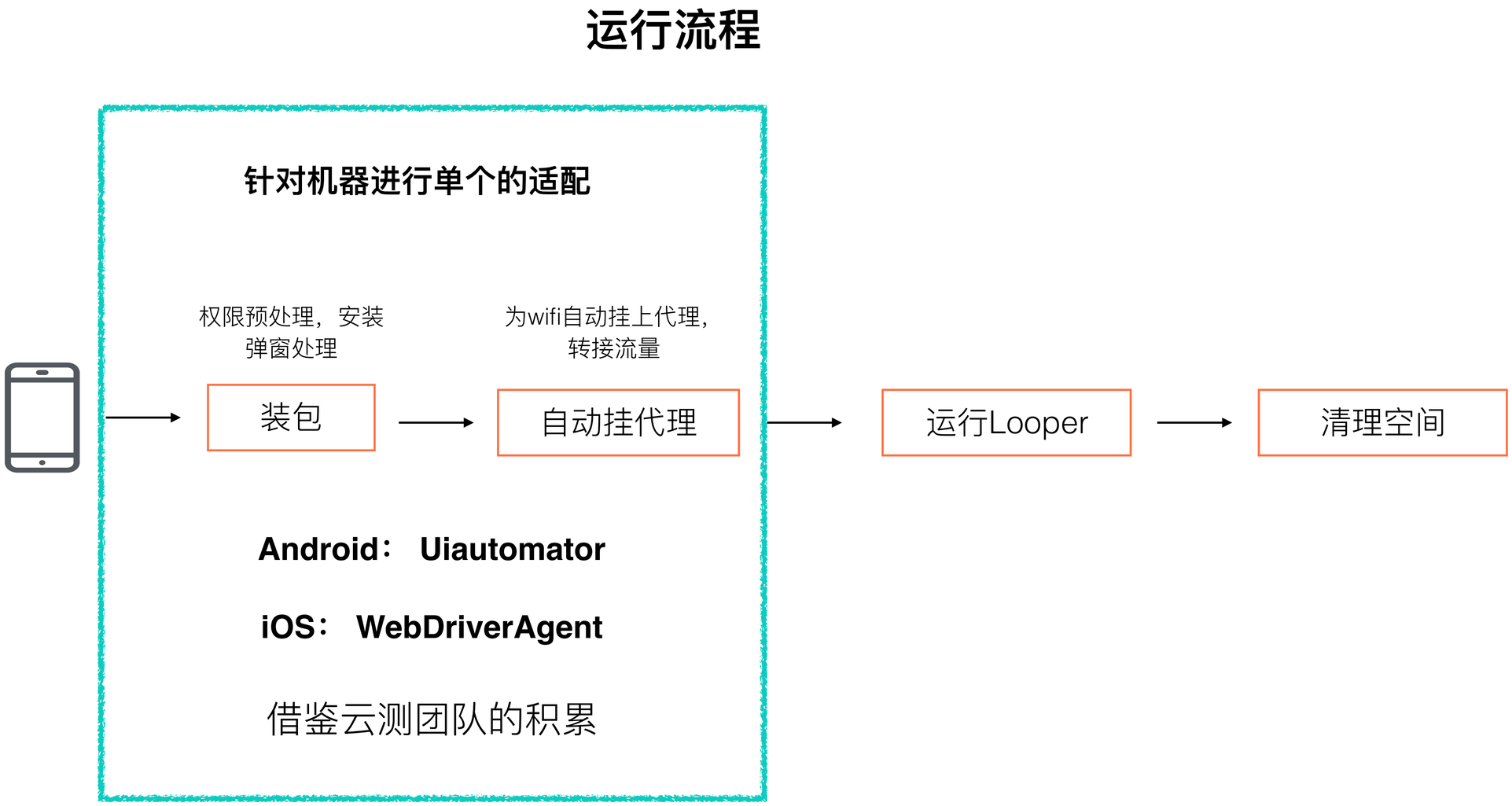
目录

有一个客户端专项工具

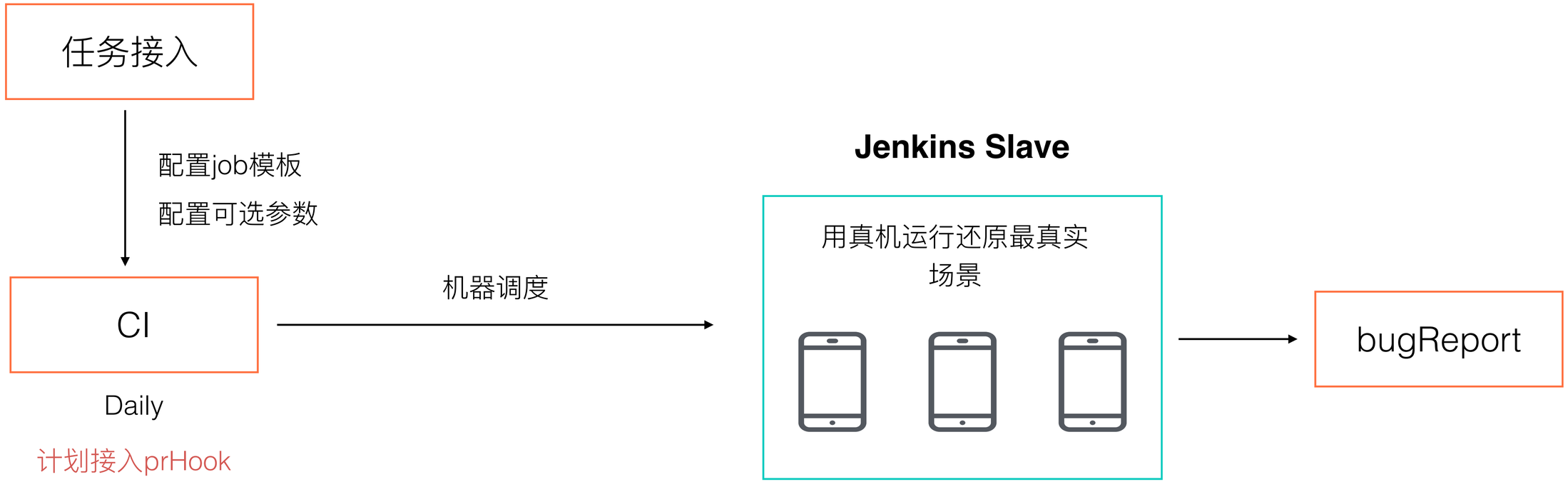
努力游

工具真正用起来发挥作用

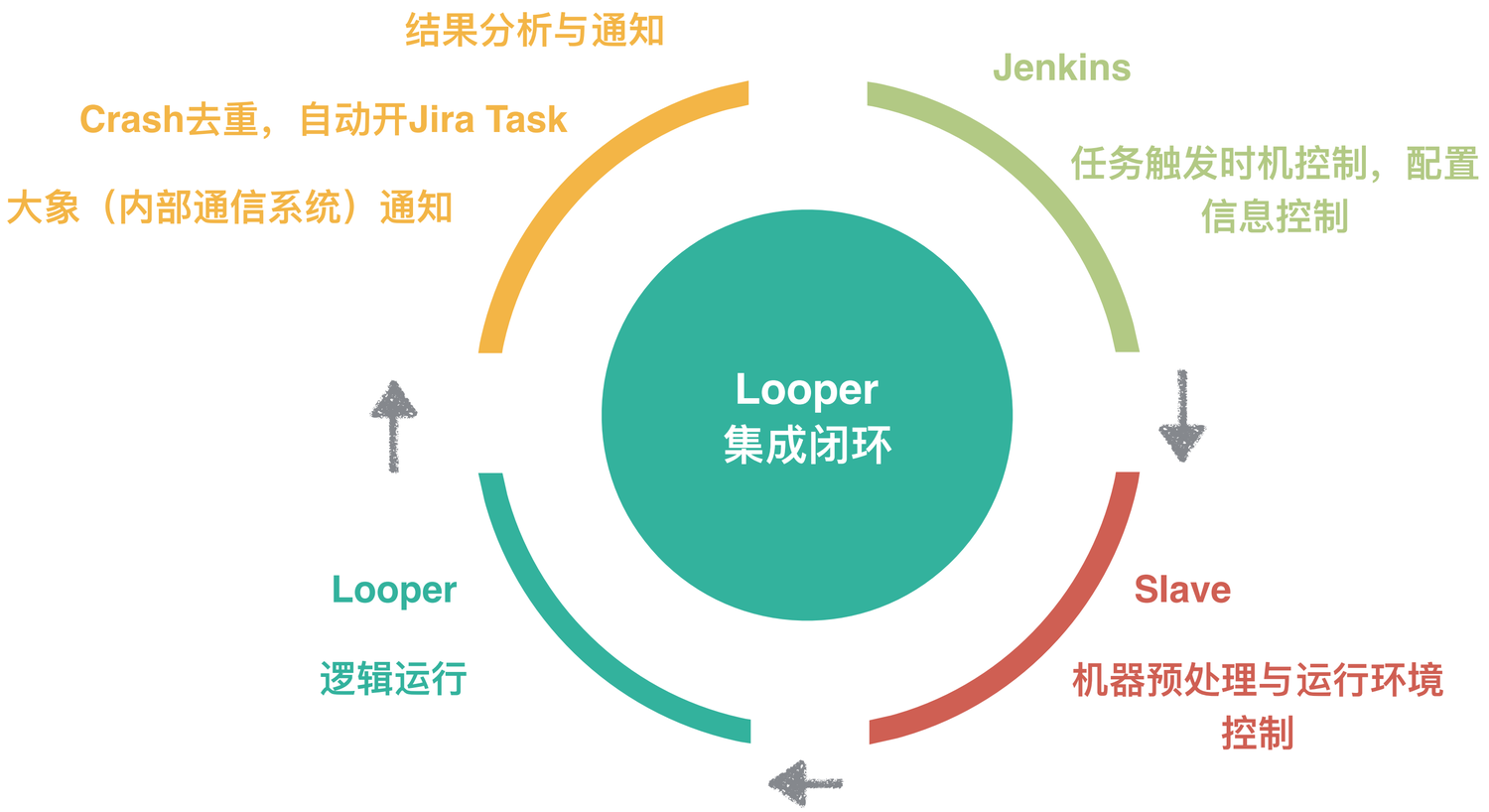
Looper单个手机运行流程



工具集成化


任务接入
配置job模板
配置可选参数
CI
Daily
计划接入prHook
机器调度
Jenkins Slave
用真机运行还原最真实场景
bugReport

工具自动集成闭环


结果分析与通知
Crash去重，自动开Jira Task
大象（内部通信系统）通知
Jenkins
任务触发时机控制，配置信息控制
Looper
集成闭环
Looper
逻辑运行
Slave
机器预处理与运行环境控制

目录

效果

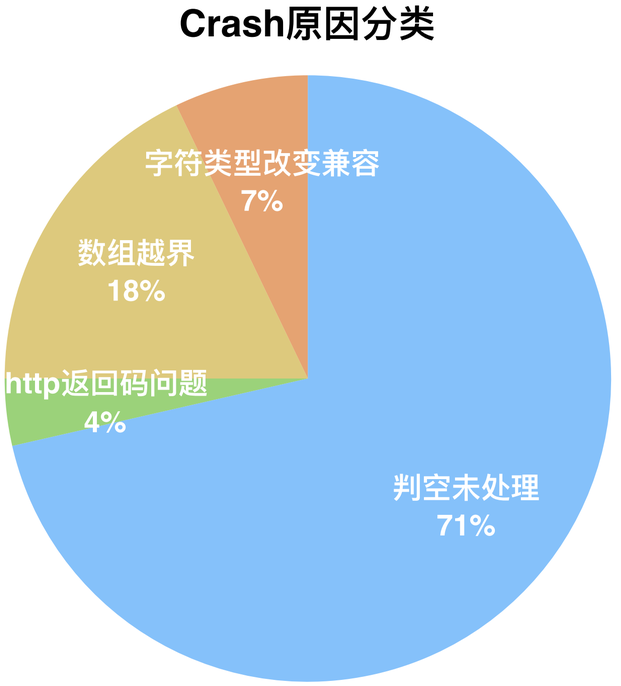
8月中旬加入集成测试运行

去重Crash 36个

- 实际给出的Crash有158个，Crash重复比率在1:5

开发修复接受率 85%

- 由于固定数据和场景可重现，bug修复转化率高，开发修复都会比较积极



展望

- 根据结果Crash分析，确定出问题大的修改组合，规划一个周期性的权重修改策略
- 通过发现的异常生成规范样例，作为开发人员在编写代码时候的准则
- 加入模拟网络条件功能，丢包，接口延迟等，模拟一些异常网络，进行接口降级监控
- 后续会进行工具的开源

Q&A

Thanks

Eat better,Live better.

美团点评

北京市朝阳区望京东路4号恒电大厦BC座

Block B&C, Hengdian Building, No.4 Wangjing East Rd,Chaoyang District, Beijing, 100102, China